DOCUMENT RESUME

ED 057 562 EM 009 399

TITLE	Project Solo; Newsletter Number Eighteen.
INSTITUTION	Pittsburgh Univ., Pa. Dept. of Computer Science.
SPONS AGENCY	National Science Foundation, Washington, D.C.
PUB DATE	7 Sep 71
NOTE	48p.; See also ED 053 566
EDRS PRICE	MF-$0.65 HC-$3.29
DESCRIPTORS	Calculus; *Computer Assisted Instruction; *Computer Programs; *Programing Languages
IDENTIFIERS	*Project Solo

ABSTRACT

An article which argues the case for extended versions of BASIC is presented in this newsletter. In addition, a booklet is attached describing a Project Solo module which is a "discovery approach" to mastering the fundamental concepts of calculus through the use of interactive computing. (JY)

PROJECT SOLO[*]

AN EXPERIMENT IN REGIONAL COMPUTING FOR SECONDARY SCHOOL SYSTEMS

University of Pittsburgh • Department of Computer Science • Pittsburgh, Pennsylvania 15213

Newsletter No. 18 September 7, 1971

Computer-Oriented Calculus

The enclosed booklet, which emphasizes a "discovery" approach to mastering the fundamental concepts of calculus through use of interactive computing, was prepared by Dr. Fred Bell for Project Solo. It actually consists of 8 modules in sequence. These modules can be used as a supplement to any standard textbook in calculus. Dr. Bell conducted some interesting evaluation with similar material at the beginning college level at Cornell University. He found a considerable improvement in understanding of concepts (e.g. meaning of a derivative) on the part of students who had computer usage over those who didn't, but little difference in mastery of fixed techniques (e.g. $d(x^2)/dx=2x$). Comments on these units by teachers and students are solicited. It should be noted that some of the material on limits would also be useful in non-calculus courses.

Com-Share News

Those of our readers who are interested in technical matters might write to Com-Share, Inc. (Box 1588, Ann Arbor, Michigan, 48106) for a copy of Vol. 4, No. 2 of the Com-Share News. In addition to the article on NEWBASIC, we would like to call your attention to the announcement of the Sigma 7 Commander II service on page 19. An opportunity to hear about selected details of this new system was given to some of us last year at Research Park in Ann Arbor; our impression then was that it will very likely be one of the most sophisticated time-sharing systems to come out of the R & D labs in some time. It is not too far out of line, in our opinion, to consider allowing students who have upgraded themselves to an appropriate level, and who have prepared justifying proposals, to access such a system. The other announcement of interest is the 60 character per second service. Such high speed would be of interest to the potential administrative users of computing in a school.

The SIGCUE Bulletin

The enclosed article presenting the case for extended versions of BASIC was prepared at the invitation of Dr. Karl Zinn for use in INTERFACE, which is the official publication of SIGCUE (the ACM Special Interest Group on Computer Uses in Education). There will be other invited articles discussing the pros and cons of this topic in the fall issue. We would recommend this publication highly to all educators. Subscription information can be obtained by writing to: SIGCUE Bulletin, 109 East Madison, Ann Arbor, Michigan, 48104.

Newsletter Renewal Form; Project Solo Film

We are cleaning up our newsletter mailing list, and request that you complete the enclosed form if you wish to receive our mailings during the coming school year. Also, a 29 minute, 16mm. color film (optical sound) on Project Solo is now available for rental. If you are interested in obtaining this film, please fill in the appropriate information on the renewal form.

*Supported in part by NSF grant GJ-1077.

The Case for Extending BASIC as an Educational Programming Language

Thomas A. Dwyer
Department of Computer Science
University of Pittsburgh

There is only one way to make something better, and that is to change it. It is important however, not to confuse the word 'change' with the word 'destroy'. On the contrary, intelligent and useful change is characterized at least as much by a concern for preserving the good aspects of that being changed, as it is with revision. It is often preferable, in fact, to view change as an augmentation rather than a drastic re-shuffling of the original system.

I believe that these general observations provide the key to answering the question "whither hence" that faces the educational community today when considering computer languages for interactive computing. The question is an important one, for there is little doubt that the use of such computing, especially in community colleges and secondary schools, is about to rise dramatically.

Having worked quite closely with secondary school administrators, teachers, and students, it has become very obvious to me that the importance of being able to communicate computer-based curricular ideas in a single "mother-tongue" within the user community cannot be overestimated. While one could argue that better languages could have been chosen for this function (my own choice at the present would be PIL-X, a fourth generation derivative of JOSS), the de-facto educational mother-tongue of BASIC turns out to be an extremely good compromise.

I do not call BASIC a compromise in a derogatory sense, but only to indicate that there is a vast spectrum of user needs even within the educational community, and that BASIC was deliberately aimed at the 'novice' end of the spectrum. This was a very wise decision indeed, and it accounts in great part for the imminent growth in educational computing referred to above. It has been our experience[1], however, that the novice stage has a very short life span for even 8^{th} and 9^{th} grade students, and that being able to capture the real potential of computing for the full span of the educational experience that lies ahead of such youngsters is critically dependent on the software we give them. Such software must incorporate a growth potential that matches the fantastic expansion in sophistication that takes place in the latter years of any person's education. For reasons that space does not permit listing, I believe that it is also desirable for the same software to accomodate other educational uses (e.g. administrative).

In order to accomodate such growth in our own work, we have gone the route of developing an expanded (and expandable) version of BASIC called NEWBASIC.[2] We originally experimented with the idea of preserving the purity of an "urtext" version of BASIC as the principle language, placing all new features (e.g. a CAI scan syntax) in a pre-processor, calling the combination NEWBASIC/CATALYST. However, at the user level, learning to use the pre-processor (we insist on user involvement) proved to be far more distracting than learning to add new language elements to one's repertoire. In practice, most of the new features were incorporated as functions, which means that the syntax of BASIC was not altered. Where the syntax was augmented, or where things like multiple statements on a

line were permitted, it was always in the spirit of serving the novice. This works well in practice. Even the youngest students understand the meaning of something like:

```
10 IF X>5  PRINT 10*X ELSE LET X=5  GOTO 100
```

To think that the beginner needs to have this spelled out for him in a "simple" form such as

```
10 IF X>5  THEN 40
20 LET X=5
30 GOTO 100
40 PRINT 10*X
```

is just as much of an educational mistake as forever speaking to youngsters with a one-syllable word vocabulary!

On the question of adding new functions, I would certainly hope for uniformity in name, parameter lists, and effect. But it seems very defeating to me to argue that nothing should be done until a standards committee designs such functions. Committees continue to be inefficient mechanisms; I would rather see them used to arbitrate differences. For example, we use a function NUM(X) which produces a truly random integer N in the range 1<=N<=X. If a standards committee could argue that it is better to call this function RAN, or that the range should include zero, fine. In the meantime, I know that we are getting more beginners involved in more interesting things because we can let them incorporate a statement like:

```
100 PRINT NUM(50) FOR I=1 TO 10
```

as part of a bigger exploration, instead of the following (which <u>is</u> required on most other systems today):

```
 10 PRINT "TYPE A NUMBER"
 20 INPUT N
 30 FOR I=1 TO N
 40 LET X=RND(Z)
 50 NEXT I
100 FOR I=1 TO 10
110 PRINT INT(50*RND(Z)+1)
120 NEXT I
```

The two codings in this last example provide a revealing contrast, analogous to that between statements in a higher level language and the corresponding code in assembly language. The reason I would always argue for the higher level version, is that it induces higher level use of computing by the man whose channel of communication with the machine <u>is</u> the language.

In sum, then, I would advocate expanded versions of BASIC[3] which also allow a simple form of extensibility by users through a true function-defining feature. The "original" form of BASIC should always be a subset of such an expanded version, and every effort should be made to standardize the form of the new features. The disadvantage of such an expanded language is that (1) the compiler rapidly becomes complex, and (2) vendors will tend to resist "compatibility" efforts. The answer to both of these difficulties is a very strong stand on the part of users, and especially funding agencies, in insisting on the primacy of the educational community as a benefactor of the best efforts of computer technologists.

---

[1] This experience is based on two years of work with "Project Solo", an experiment concerned with the use of interactive computing in secondary schools. A newsletter, sample curriculum modules, and other documentation on the project are available from the author. A film documenting the first year of the project is also available for rental (16mm. color, 29 minutes, optical sound).

[2] The NEWBASIC system was developed in cooperation with Com-Share, Inc., a commercial time-sharing company based in Ann Arbor, Michigan. One advantage to involving a commercial time-sharing vendor in an educational project is the capability for immediate transportability to a large number of cities (55 in this case) via their continental network, as well as abroad.

[3] I also advocate expanding user interaction capabilities, but this is a system-design rather than language-design consideration. The total NEWBASIC system incorporates a number of such capabilities.

Project Solo Newsletter Subscription Request Form - September 1971

Please continue to send 1 copy of the Project Solo Newsletter along with selected modules to:

NAME: ______________________

INSTITUTION: ______________________

ADDRESS: ______________________

______________________ ZIP: ______

Reader Comments

1. Material you like most, or would want more of:

2. Things you would like to see added (or deleted) to the mailings:

3. Other:

Film Request

The undersigned is interested in renting the Project Solo film (1 week), to be shown to audience(s) totalling ____ people. Please send rental information to: NAME: ______________________

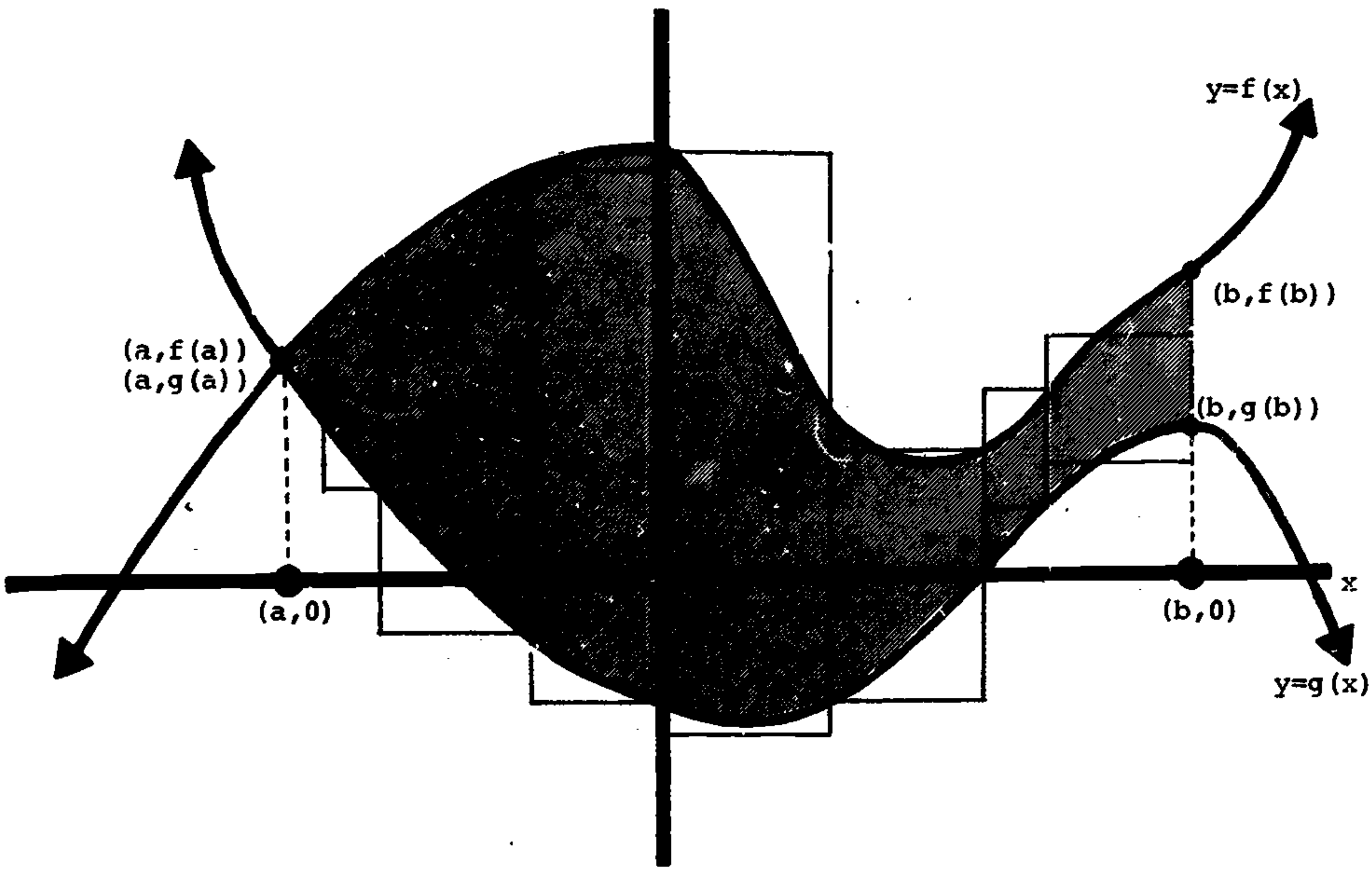
INSTITUTION: ______________________

ADDRESS: ______________________

______________________ ZIP: ______

computer-augmented calculus topics

PROJECT SOLO
Department of Computer Science
University of Pittsburgh (15213)
Modules 0017, 0019, 0031, 0049, 0094, 0104, 0106, 0122.



SOME HINTS ON THE LIMITATIONS OF COMPUTERS FOR CALCULATIONS

When you are solving some of the problems in this booklet you may encounter difficulties that appear to be caused by inconsistencies in mathematical methods but that are actually caused by the manner in which computers store and process numbers. Computers can not handle numbers which have too many significant digits or which are too large in absolute value. In NEWBASIC you are limited to numbers having ten significant digits that range in absolute value from 10^{-76} to 10^{+76}. Problems also may arise when computers try to divide by numbers close to zero, subtract two numbers which are very close together, or add a very small decimal number to a very large whole number. One of the jobs which many mathematicians, computer scientists, and engineers work on is that of devising ingenious schemes for getting around these limitations of computers.

The following examples show a few of the kinds of mathematical problems that will be solved incorrectly by computers.

Example 1: The problem of finding 2^{793} can not be solved directly on a computer because 2^{793} is too large in absolute value to be calculated by an instruction such as >10 LET Y=2.0↑793. This statement will cause the computer to print out X IN EXP(X) TOO LARGE LINE 10.

Example 2: $8 + 10^{-13}$ is 8.0000000000001, but the instructions:

```
>10 LET X=8.0
>20 LET Y=1.0E-13
>30 LET Z=X + Y
>40 PRINT Z
>50 END
```

will give the following printout for Z.

8

Example 3: The problem here is to find $\frac{x}{\sqrt{x+9}-3}$ for x=.000000001.

The program:

```
>10 LET X=.000000001
>20 LET Y=X/(SQRT(X+9.0)-3.0)
>30 PRINT Y
>40 END
```

gives 6.247225158 for the value of Y and this is not the correct solution to the problem.

However, we can use the computer to solve this problem if first we rationalize the denominator.

$$\frac{x}{\sqrt{x+9}-3} = \sqrt{x+9}+3$$

The program:

```
>10 LET X=.000000001
>20 LET Y=SQRT(X+9.0) +3.0
>30 PRINT Y
>40 END
```

gives the correct answer of 6 as output.

LIMITS OF SEQUENCES, LIMITS OF SUMS OF SEQUENCES, LIMITS OF A FUNCTION

The speed and accuracy of computers does make them particularly usefull for exploring limiting processes in mathematics. For example the sequence defined by $f(n)=\sqrt[n]{n}$ (for n=1,2,3,···) is made up of terms close to 1 for large values of n. The computation of a sufficient number of terms having large enough indices to enable us to be reasonably certain that $\lim_{n\to\infty} \sqrt[n]{n} = 1$ is quite tedious if we use pencil and and paper; however, the simple program:

```
>10 INPUT N
>20 LET F=N↑(1.0/N)
>30 PRINT N, F
>40 GOTO 10
```

permits us to compute enough terms of the sequence to discover that 1 is a good guess for the limit. The following table shows that this sequence converges to its limit rather slowly.

N	F	N	F	N	F	N	F
1	1.000	6	1.348	20	1.162	200	1.027
2	1.414	7	1.320	30	1.120	300	1.019
3	1.442	8	1.297	40	1.097	400	1.015
4	1.414	9	1.277	50	1.081	500	1.013
5	1.380	10	1.259	100	1.047	1000	1.007

Using a computer to "make a sequence approach its limit" can help to clarify our understanding of limits; however, we must exercise extreme caution in drawing conclusions. Since we can never look at all of the terms, any conclusion about the limit of the sequence must be considered at best an educated guess and is always suspect until we can find some way to prove our guess.

Problem 1: Write programs and use them to help you to make guesses about the limits of these sequences.

A. $\lim_{n\to\infty} \sqrt[n]{10}$

B. $\lim_{n\to\infty} \frac{2n+10}{n}$

C. $\lim_{n\to\infty} \frac{1}{2^n}$

D. $\lim_{n\to\infty} \frac{n^2}{2^n}$

(Be careful, for large values of n, 2^n is too large for the computer.)

E. $\lim_{n\to\infty} \frac{n+1}{n-1}$ (What happens if n=1?)

F. $\lim_{n\to\infty} \left(\frac{2}{3}\right)^n$

G. $\lim_{n\to\infty} \left(1+\frac{1}{n}\right)^n$ (This function has a limit, but for very large n the computer gives incorrect values for $\left(1+\frac{1}{n}\right)^n$.

Problem 2: For each real number p, $f(n)=\frac{1}{n^p}$ determines a sequence for $n=1,2,3,\cdots$. For some values of n the sequence converges and for others it deverges. Write a program that can be used to assist us in finding $\lim_{n\to\infty} \frac{1}{n^p}$ for any value of p. Use your program to find the values of p for which this sequence will converge.

Problem 3: Make up some interesting sequences and write programs to help you discover their limits.

The problem of finding the limit of the sum of the terms of a sequence can also be approached with the assistance of a computer. For instance the following program will sum the first n terms of the sequence 1, 1/2, 1/4, 1/8, $\cdots$ for any integral value of n.

```
>10 LET S=1
>20 INPUT N
>30 FOR I=1 TO N
>40 LET F=1.0/(2.0↑I)
>50 LET S=S+F
>60 PRINT I, S
>70 NEXT I
>80 END
```

For n=10 we have:

I	S	I	S
1	1.5	6	1.984375
2	1.75	7	1.9921875
3	1.875	8	1.99609375
4	1.9375	9	1.998046875
5	1.96875	10	1.999023437

From this table it appears that the limit may be 2.

Problem 4: Write programs and use them to estimate the limits of the sums of these sequences:

(A) $1, 1/3, 1/9, 1/27, \cdots$

(B) $(1/2)^2, (1/3)^2, (1/4)^2, \cdots$

(C) $\frac{1}{0!}, \frac{1}{1!}, \frac{1}{2!}, \frac{1}{3!}, \cdots$

(D) $1, -1/2, 1/3, -1/4, 1/5, \cdots$

(E) $1, 1/2, 1/3, 1/4, 1/5, \cdots$

We can build upon our procedures for exploring limits of sequences and limits of sums of sequences to study the limit of a function defined on any subset of the real numbers.

For example, consider $\lim_{x \to 1} \frac{x^3+x^2-x-1}{x-1}$. Since x can have any real value except 1, we must decide upon a way to let x approach 1 in order to obtain useful data about this limit. One of the many possibilities is shown next.

This program with N=5

```
10 INPUT N
15 PR.
20 FOR I=1 TO N
30 LET D=10↑(-I)
40 LET X1=1-D
50 LET X2=1+D
60 LET Y1=(X1↑3+X1↑2-X1-1)/(X1-1)
70 LET Y2=(X2↑3+X2↑2-X2-1)/(X2-1)
```

```
 80 PR. X1,Y1
 90 PR. X2,Y2
100 NEXT I
110 END
```

gives this output.

0.9	3.61
1.1	4.41
0.99	3.960100001
1.01	4.040099998
0.999	3.99600101
1.001	4.004000991
0.9999	3.999600113
1.0001	4.000399887
0.99999	3.99996071
1.00001	4.00003929

The output indicates (but certainly doesn't prove) that the limit may be 4.

You can see that the programming skills needed to write programs for exploring limits of sequences and functions are minimal and even people with little experience on the terminal can soon be exploring some rather sophisticated mathematics concepts.

<u>Problem 5</u>: Write programs and use them to make an educated guess about the following limits:

(A) $\lim_{x \to 0} \left(\frac{1}{1+x}\right)^x$

(B) $\lim_{x \to 0} \frac{\sin(x)}{x}$

(C) $\lim_{x \to 1} \frac{x^2-1}{x-1}$

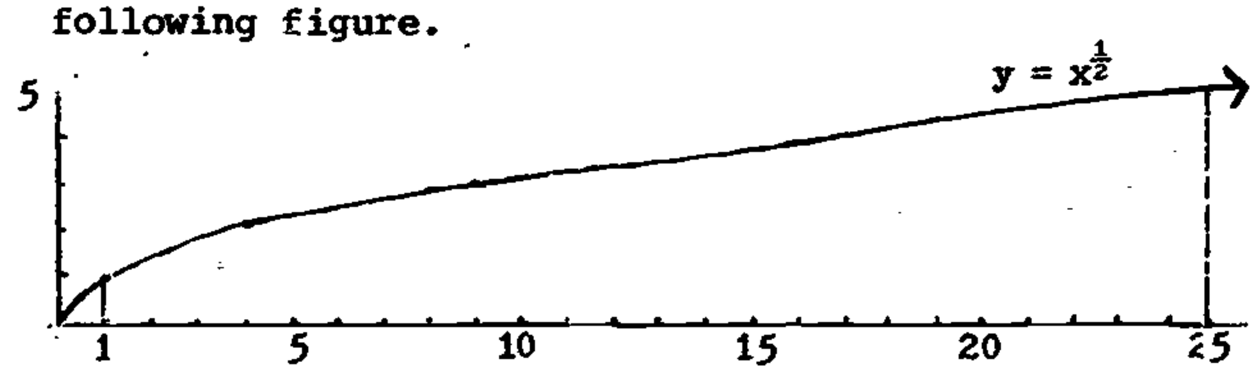
(D) $\lim_{x \to 0} \frac{x}{\sqrt{x+4}-2}$ (See Example 3)

(E) $\lim_{x \to 0} (1+x)^{1/x}$ (You may run into a problem here.)

(F) Compare Problem 5, part E to Problem 1, part G.

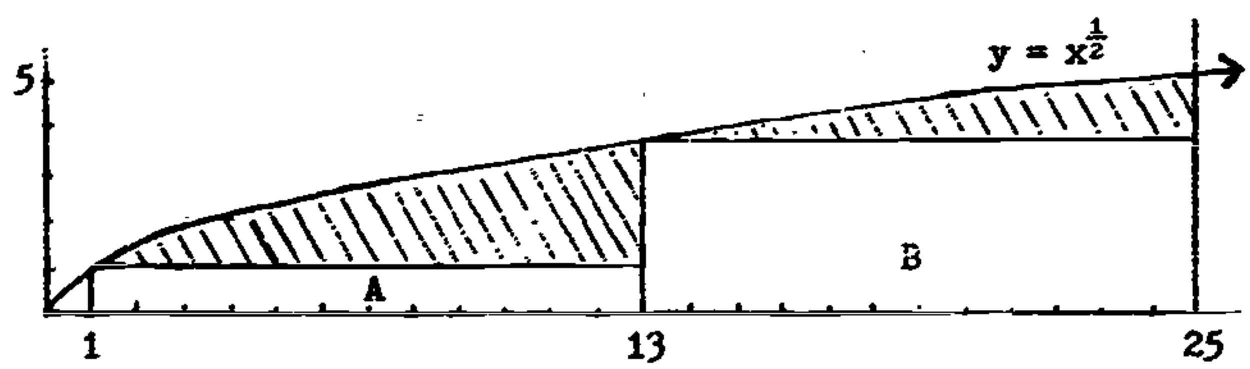
APPROXIMATING AREAS

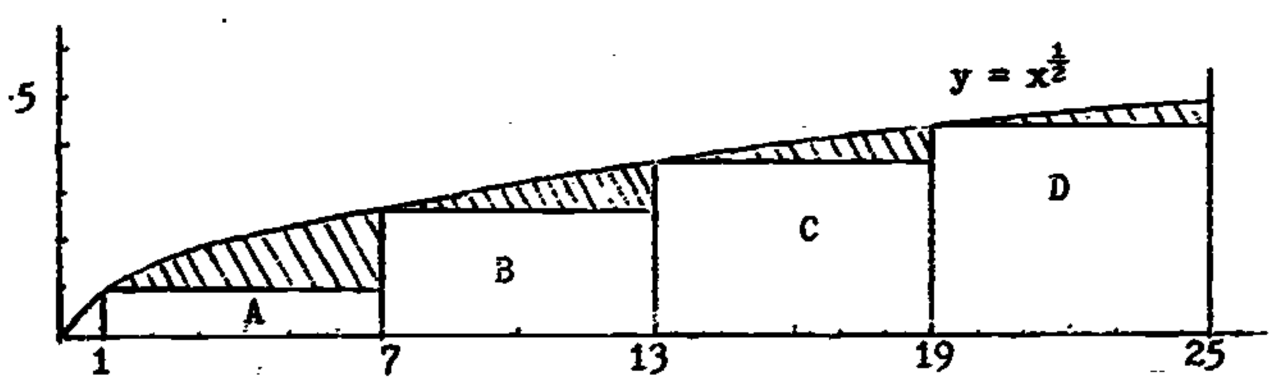
Our problem now is that of finding approximations to areas bounded by the graph of a function and an interval of the x-axis. For example suppose we want to find the area bounded by the graph of $y=x^{\frac{1}{2}}$ and the interval [1, 25] of the x-axis. This area is illustrated in the following figure.

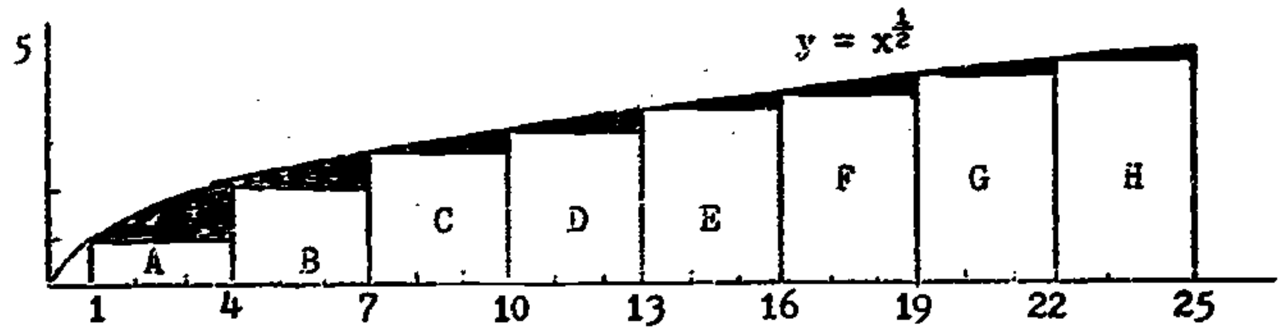


The first approach we will use to approximate this area is that of constructing rectangles, the sum of whose areas is an approximation to the area in question.

The next set of diagrams illustrates ways of building rectangles to get increasingly more accurate approximations.





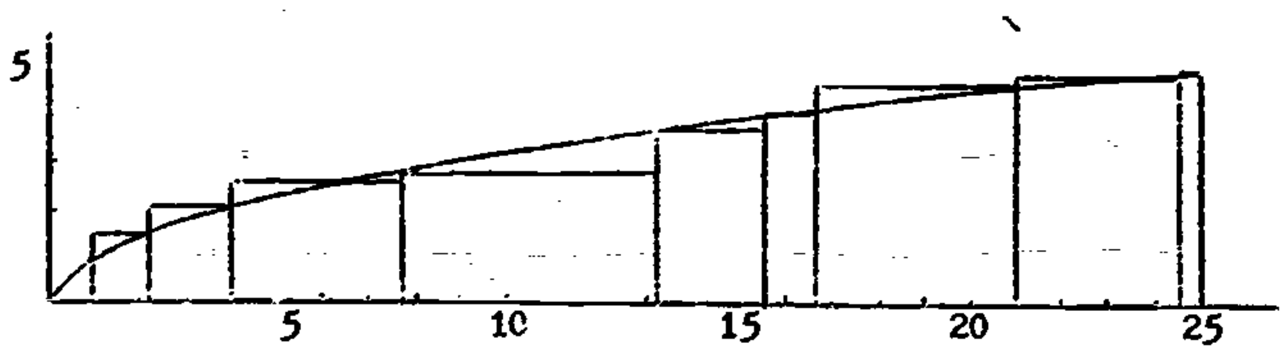


With two rectangles the approximation is $12(1)^{\frac{1}{2}} + 12(13)^{\frac{1}{2}} = 53.27$. With four rectangles it is $6(1)^{\frac{1}{2}} + 6(7)^{\frac{1}{2}} + 6(13)^{\frac{1}{2}} + 6(19)^{\frac{1}{2}} = 69.66$. With eight rectangles our approximation is 76.39.

You can observe from the shaded areas in the figures that the difference between the area and our approximation (the error) decreases as we increase the number of rectangles. The procedure of doubling the number of rectangles was continued to 8192 rectangles to obtain the following table.

Number of rectangles	Approximation to the area	Number of rectangles	Approximation to the area
2	55.266	128	82.291
4	69.661	256	82.479
8	76.388	512	82.573
16	79.594	1024	82.620
32	81.148	2048	82.643
64	81.912	4096	82.655
		8192	82.661

We could have constructed some of the rectangles to extend above the graph and the widths of their bases could have been varied as shown below.



The next program /AREA/ is an example of the "rectangle method" (Eulers method) for obtaining a set of approximations to the area bounded by the graph of any specified polynomial of degree three or less for any interval of the x-axis. When you use the program you are free to select the polynomial and the interval. After the program has calculated an approximation to the area bounded by $ax^3 + bx^2 + cx + d$ over $[r,s]$ four options are available to you:

1. Use more rectangles to get a better approximation.
2. Approximate an area bounded by the same function over a different interval.
3. Approximate the area bounded by a different function.
4. Quit and do something else.

When using the program, after you get the message NOW TYPE EITHER 34, 44, 54, OR 1000

1. Type 54 for option 1.
2. Type 44 for option 2.
3. Type 34 for option 3.
4. Type 1000 for option 4.

A sample run together with the program listing is shown on the next two pages.

A sample run of the program /AREA/.

```
THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE AREA
BØUNDED BY THE X-AXIS AND THE GRAPH ØF ANY PØLYNØMIAL
FUNCTIØN ØF DEGREE THREE ØR LESS ØVER AN INTERVAL ØF
THE X-AXIS.  THE METHØD USED TØ APPRØXIMATE THE AREA
IS TØ BUILD RECTANGLES WHØSE SUM WILL BE AN APPRØXIMATIØN
TØ THE AREA WE ARE TRYING TØ FIND.
THE PØLYNØMIAL FUNCTIØN IS ØF THE FØRM

A*X↑3 + B*X↑2 + C*X + D

YØU CAN GIVE ANY VALUES (INCLUDING D) TØ A,B,C, AND D.
Ø.K., ASSIGN NUMERICAL VALUES TØ A, B, C, AND D.
A =
?D
B =
?1
C =
?D
D =
?D
```

/AREA/ (continued)

```
SELECT AN INTERVAL [R,S] ØN THE X-AXIS
BY ASSIGNING VALUES TØ R AND S.
R =
?D
S =
?2
HØW MANY RECTANGLES DØ YØU WANT TØ BUILD ØN [R,S]?
N =
?4


THE INTERVAL IS [     D     ,    2    ].
THE NUMBER ØF RECTANGLES IS    4    .
THE FUNCTIØN IS
 D    X↑3 +     1    X↑2 +     D    X +     D    .

THE APPRØXIMATIØN TØ THE AREA IS *****     1.75     *****


NØW TYPE EITHER 34, 44, 54, ØR 1DDD.
?54
HØW MANY RECTANGLES DØ YØU WANT TØ BUILD ØN [R,S]?
N =
?8


THE INTERVAL IS [     D     ,    2    ].
THE NUMBER ØF RECTANGLES IS    8    .
THE FUNCTIØN IS
 D    X↑3 +     1    X↑2 +     D    X +     D    .

THE APPRØXIMATIØN TØ THE AREA IS *****     2.1875   *****


NØW TYPE EITHER 34, 44, 54, ØR 1DDD.
?54
HØW MANY RECTANGLES DØ YØU WANT TØ BUILD ØN [R,S]?
N =
?64


THE INTERVAL IS [     D     ,    2    ].
THE NUMBER ØF RECTANGLES IS    64   .
THE FUNCTIØN IS
 D    X↑3 +     1    X↑2 +     D    X +     D    .

THE APPRØXIMATIØN TØ THE AREA IS *****     2.6D4492188    *****


NØW TYPE EITHER 34, 44, 54, ØR 1DDD.
```

A listing of the program /AREA/.

```
10 PR. "THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE AREA"
12 PR. "BØUNDED BY THE X-AXIS AND THE GRAPH ØF ANY PØLYNØMIAL"
14 PR. "FUNCTIØN ØF DEGREE THREE ØR LESS ØVER AN INTERVAL ØF"
16 PR. "THE X-AXIS.  THE METHØD USED TØ APPRØXIMATE THE AREA"
18 PR. "IS TØ BUILD RECTANGLES WHØSE SUM WILL BE AN APPRØXIMATIØN"
20 PR. "TØ THE AREA WE ARE TRYING TØ FIND."
24 PR. "THE PØLYNØMIAL FUNCTIØN IS ØF THE FØRM"
26 PR.
28 PR. "A*X↑3 + B*X↑2 + C*X + D"
30 PR.
32 PR. "YØU CAN GIVE ANY VALUES (INCLUDING 0) TØ A,B,C, AND D."
34 PR. "Ø.K., ASSIGN NUMERICAL VALUES TØ A, B, C, AND D."
36 PR. "A = " INPUT A
38 PR. "B = " INPUT B
40 PR. "C = " INPUT C
42 PR. "D = " INPUT D
44 PR. "SELECT AN INTERVAL [R,S] ØN THE X-AXIS"
46 PR. "BY ASSIGNING VALUES TØ R AND S."
48 PR. "R = " INPUT R
50 PR. "S = " INPUT S
52 IF R>=S GØTØ 500
54 PR. "HØW MANY RECTANGLES DØ YØU WANT TØ BUILD ØN [R,S]?"
56 PR. "N = " INPUT N
57 PR.
58 PR.
60 LET P=INT(N)
62 IF (N<1) ØR (N<>P) ØR (N>500) GØTØ 600
64 LET T = 0
66 LET W = (S-R)/N
70 FØR X = R TØ (S-W) STEP W
72 LET T = T + ABS(A*X↑3+B*X↑2 + C*X + D)
74 NEXT X
76 LET T = T*W
78 PR. "THE INTERVAL IS [";R;",";S;"]."
80 PR. "THE NUMBER ØF RECTANGLES IS ";N;"."
82 PR. "THE FUNCTIØN IS"
84 PR. A;"X↑3 + ";B;"X↑2 + ";C;"X + ";D;"."
86 PR.
88 PR. "THE APPRØXIMATIØN TØ THE AREA IS *****";T;"*****"
89 PR.
90 PR.
104 PR. "NØW TYPE EITHER 34, 44, 54, ØR 1000."
106 INPUT K
108 IF (K<>34) AND (K<>44) AND (K<>1000) AND (K<>54) GØTØ 700
110 IF K=34 GØTØ 34
111 IF K=44 GØTØ 44
112 IF K=54 GØTØ 54
113 IF K=1000 GØTØ 1000
500 PR. "R MUST BE LESS THAN S. TRY AGAIN."
505 GØTØ 44
600 PR. "N MUST BE A PØSITIVE WHØLE NUMBER "
605 PR. "LESS THAN 500.  TRY AGAIN."
610 GØTØ 54
700 PR. "YØU GØØFED.  TAKE IT FRØM THE TØP."
710 GØTØ 104
1000 END
```

Problem 6: Use the program /AREA/ to find good approximations to the following areas:

A. x^2 over [0, 4]
B. x^2 over [-4, 4]
C. x^3 over [-2, 0]
D. x^3 over [0, 2]
E. x^3 over [-2, 2]
F. $2x^3-3x^2+x-1$ over [-4, 3]

Problem 7: Write a program and use it to approximate the area bounded by sin(x) over the intervals.

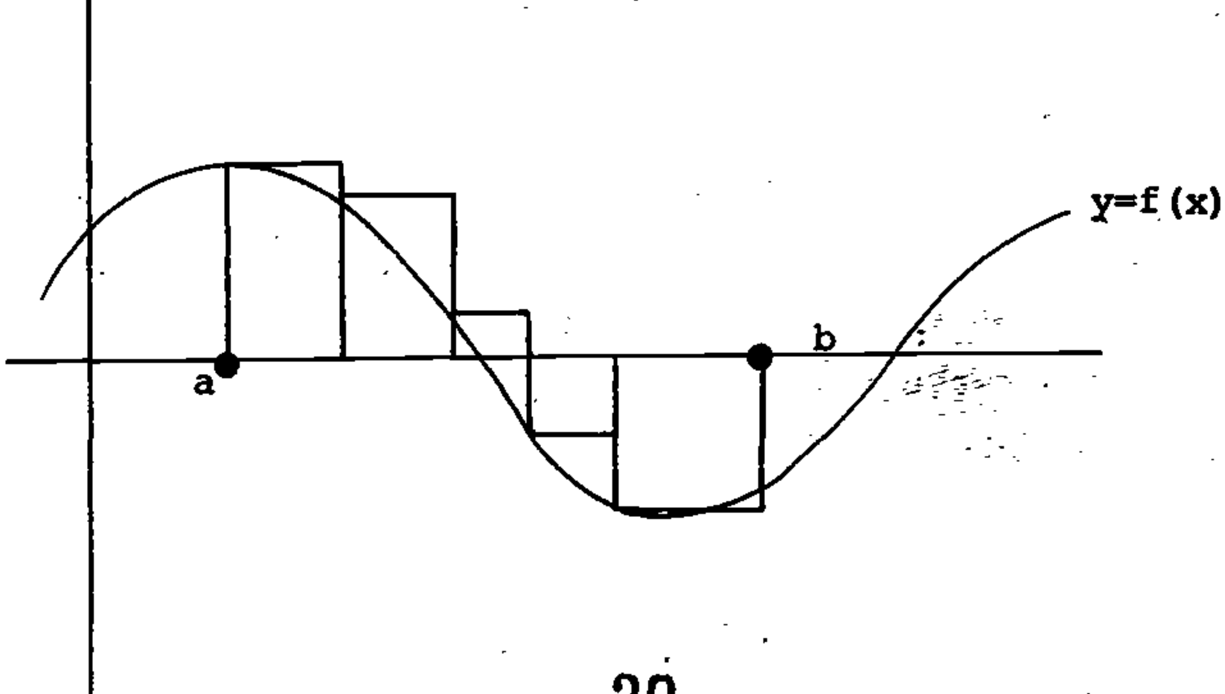
A. $[-\pi, 0]$
B. $[0, \pi]$
C. $[-\pi, \pi]$

NUMERICAL INTEGRATION

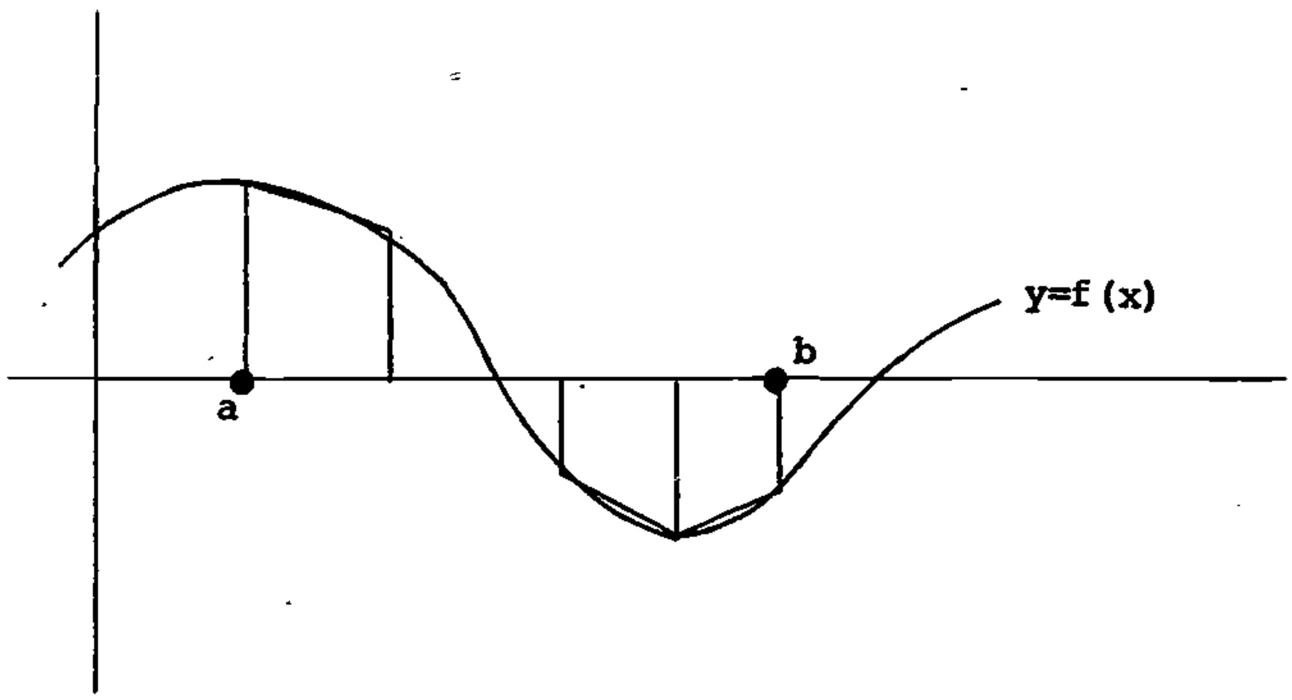
The method of constructing rectangles to approximate areas is one of several strategies which can be used in numerical integration. Remember that we can define the integral $\int_a^b f(x)dx$ to be $\lim_{\Delta x_i \to 0} \sum_{i=1}^{n} (\Delta x_i) f(x_i)$, where the Δx_i's are subintervals of the x-axis which partition the interval [a, b] and the $f(x_i)$'s are respective points in these intervals. To obtain a set of approximations to $\int_a^b f(x)dx$, we need to compute $\sum_{i=1}^{n} (\Delta x_i) f(x_i)$ where n, the number of subintervals into which [a, b] is divided, is increased for each approximation and the Δx_i's are decreased. In effect we are constructing rectangles of base Δx_i with height $f(x_i)$, for i=1, 2, 3, 4,$\cdots$, and summing the areas of all the rectangles. Since some of the $f(x_i)$'s could be negative, which occurs over subintervals where the graph of f(x) dips below the x-axis, some of our areas $(\Delta x_i) f(x_i)$ will be negative.

The next program /INTEGRATE/ is an illustration of the trapezoid method for approximating the definite integral of some function f(x). In this method we use approximating trapezoids rather than approximating rectangles as is shown in the diagram.

An approximation to $\int_a^b f(x)dx$ using rectangles.



An approximation to $\int_a^b f(x)dx$ using trapezoids.



After you have loaded the program /INTEGRATE/ do <u>not</u> type >RUN. You must alter the program by supplying it with the function whose definite integral you want to approximate. This function must be entered as line 100 of the program. For example if you want to integrate the function $y=x^2 + 2x - 1$ over some interval, you must type

```
>100 LET Y=X↑2 + 2.0*X -1.0.
```

Now the program is complete and you can use it by typing >RUN. The dependent variable must be called Y and the independent variable X; so line 100 should always have the form

```
>100 LET Y=F(X)
```

where F(X) is the function of X which you are going to study.

After obtaining an approximation to $\int_a^b f(x)dx$ you have three choices when the program calls for the change code by typing the message

TYPE THE CHANGE CODE.

Choice 1: Type 30 to get a better approximation by increasing the number of trapezoids.
Choice 2: Type 18 to integrate the same function over a different interval.
Choice 3: Type 300 to stop the program.

If you type 300 you can either stop or try the program on a different function Y=G(X) by typing

```
>100 LET Y=G(X)
>RUN
```

Problem 8: Use /INTEGRATE/ to obtain a set of approximations to $\int_0^2 e^x dx$. Make a guess as to the value of this integral.

Problem 9: Use integrate to approximate:

A. $\int_1^2 \log_{10}(x)dx$

B. $\int_2^4 \log_{10}(x)dx$

C. $\int_1^4 \log_{10}(x)dx$

Problem 10: Use /INTEGRATE/ to approximate $\int_0^1 \log_2(x)dx$. Some ingenuity is required here because $\log_2(0)$ is undefined.

Problem 11: Sketch a graph of $y=x^3-9x$. Use /INTEGRATE/ to approximate the following definite integrals.

A. $\int_{-4}^{3} (x^3-9x)dx$

B. $\int_{-3}^{0} (x^3-9x)dx$

C. $\int_{0}^{3} (x^3-9x)dx$

D. $\int_{-4}^{-3} (x^3-9x)dx$

Interpret the results of A, B, C, D relative to your graph.

A sample run of the program /INTEGRATE/.

```
>100 LET Y = X↑2 - 2.0*X
>RUN
THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE
DEFINITE INTEGRAL ØF THE FUNCTIØN WHICH YØU
SUPPLIED ØN LINE 100, ØVER AN INTERVAL [A,B].

NØW TYPE A VALUE FØR A.
?1
TYPE THE VALUE ØF B.
?3
HØW MANY SUBINTERVALS DØ YØU WANT [A,B] DIVIDED INTØ?
?16


THE INTERVAL IS [      1     ,    3    ].
THE NUMBER ØF APPRØXIMATING TRAPEZØIDS IS      16

THE APPRØXIMATIØN IS  ******   0.671875    ******

TYPE THE CHANGE CØDE.
?30
HØW MANY SUBINTERVALS DØ YØU WANT [A,B] DIVIDED INTØ?
?100


THE INTERVAL IS [      1     ,    3    ].
THE NUMBER ØF APPRØXIMATING TRAPEZØIDS IS      100

THE APPRØXIMATIØN IS  ******   0.696799998   ******

TYPE THE CHANGE CØDE.
?30
HØW MANY SUBINTERVALS DØ YØU WANT [A,B] DIVIDED INTØ?
?400


THE INTERVAL IS [      1     ,    3    ].
THE NUMBER ØF APPRØXIMATING TRAPEZØIDS IS      400

THE APPRØXIMATIØN IS  ******   0.674174994   ******

TYPE THE CHANGE CØDE.
?500
THE CHANGE CØDE MUST BE 18,30, ØR 300.
TYPE THE CHANGE CØDE.
?30
HØW MANY SUBINTERVALS DØ YØU WANT [A,B] DIVIDED INTØ?
?500


THE INTERVAL IS [      1     ,    3    ].
THE NUMBER ØF APPRØXIMATING TRAPEZØIDS IS      500

THE APPRØXIMATIØN IS  ******   0.672671988   ******
TYPE THE CHANGE CØDE.
?
 300
```

A listing of the program /INTEGRATE/.

```
10 PR. "THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE"
12 PR. "DEFINITE INTEGRAL ØF THE FUNCTIØN WHICH YØU"
14 PR. "SUPPLIED ØN LINE 100, ØVER AN INTERVAL [A,B]."
16 PR.
18 PR. "NØW TYPE A VALUE FØR A." INPUT A
20 PR. "TYPE THE VALUE ØF B." INPUT B
25 IF A>=B GØTØ 200
30 PR. "HØW MANY SUBINTERVALS DØ YØU WANT [A,B] DIVIDED INTØ?"
35 INPUT N
40 LET N=FIX(ABS(N))
45 LET S=0
50 LET W=(B-A)/N
55 FØR X=A TØ B STEP W
100 LET Y = X↑2 - 2*X
105 IF X = B GØTØ 125
110 IF X = A GØTØ 125
115 LET S=S + 2*Y
120 GØTØ 130
125 LET S = S + Y
130 NEXT X
135 LET T = ((B-A)/(2*N))*S
140 PR.
142 PR.
144 PR. "THE INTERVAL IS [";A;",";B;"]."
146 PR. "THE NUMBER ØF APPRØXIMATING TRAPEZØIDS IS";N
148 PR.
150 PR. "THE APPRØXIMATIØN IS  ******";T;"******"
152 PR.
155 PR. "TYPE THE CHANGE CØDE." INPUT C
160 IF C=18 GØTØ 18
165 IF C=30 GØTØ 30
170 IF C=300 GØTØ 300
175 GØTØ 250
200 PR."A MUST BE LESS THAN B. TRY AGAIN."
205 GØTØ 18
250 PR. "THE CHANGE CØDE MUST BE 18,30, ØR 300."
255 GØTØ 155
300 END
```

APPROXIMATING VALUES OF DERIVATIVES

The derivative of f(x) at x=a can be defined as

$\lim_{h\to 0} \frac{f(a+h)-f(a)}{h}$. Since a is a constant, $\frac{f(a+h)-f(a)}{h}$

is a function of h; consequently any method developed for approximating the limit of a function can be used to approximate this limit.

To approximate the derivative of $y=x^2$ for x=3 directly from definition, we need to find values of $\frac{(3+h)^2-(3)^2}{h}$ for values of h which are close to 0. The program shown below produces the following set of approximations to $\lim_{h\to 0} \frac{(3+h)^2-(3)^2}{h}$, where N=5. Notice that in this program h is approaching 0 from the positive side.

```
10 INPUT N
20 FOR I=1 TO N
30 LET H=1/(10↑N)
40 LET D=(((3+H)*2)-(3*2))/H
50 PRINT H, D
60 NEXT I
70 END
```

h	$\frac{(3+h)^2-(3)^2}{h}$
.1	6.099999999
.01	6.009999994
.001	6.000999885
.0001	6.000099238
.00001	5.999999121

Problem 12: Write a program which will compute values of $\frac{(3+h)^2-(3)^2}{h}$ for negative values of h. Use your program to estimate $\lim_{h\to 0} \frac{(3+h)^2-(3)^2}{h}$ as h approaches 0 from the negative side. What is the derivative of x^2 when x = 3?

The problem of finding $\frac{(3+h)^2-(3)^2}{h}$ for values of h which are close to 0 illustrates one of the difficulties that can occur when we use a computer to perform our calculations. Suppose h=.00000000001, then the computer must find $\frac{(3+.00000000001)^2-3^2}{.00000000001}$. However, 3+.00000000001 = 3.00000000001, which has 12 significant digits, and our computer can't handle this many significant digits, consequently the computer will take 3.00000000001 to be 3.0000000000 and compute $\frac{(3.0000000000)^2-(3.0000000000)^2}{1.0E-11}$ which is 0. The correct answer is 6.00000000001, and in fact we are going to get erroneous data when $|h|<.0001$. We can avoid this problem by simplifying $\frac{(3+h)^2-(3)^2}{h}$ before we write our computer program.

$$\frac{(3+h)^2-(3)^2}{h} = \frac{9+6h+h^2-9}{h} = 6+h$$

So, $\lim_{h\to 0} \frac{(3+h)^2-(3)^2}{h} = \lim_{h\to 0} (6+h)$, and for h close to 0, 6+h is close to 6.

If we don't make this simplification the computer gives us incorrect information. If we make the simplification we don't really need a computer to show us that $\lim_{h\to 0} (6+h) = 6$. In this problem the computer really isn't of much help.

As a general rule when using the computer to find $\frac{f(x+h)-f(x)}{h}$ for various values of x, you should avoid letting h be closer than .00001 to 0 as you attempt to estimate $\lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$.

Problem 13: Write a program which can be used to find estimates for the derivative of $\log_{10} x$ for any value of x which the user of the program chooses. You will need to consider finding $\lim_{h\to 0} \frac{\log_{10}(x+h)-\log_{10}(x)}{h}$.

Remember that $\log_{10}x$ is only defined for positive values of x. Test your program by estimating the derivatives for x=10, 100, 1, 2, and 5.

The next program called /DERIVSIN/ is one which can be used to find the derivative function of sin(x). Our strategy will be to find approximations to $\lim_{h\to 0} \frac{\sin(x+h)-\sin(x)}{h}$ for many values of x, obtaining a set of pairs of numbers $\{(x, \text{approximation to} \lim_{h\to 0} \frac{\sin(x+h)-\sin(x)}{h})\}$ for these values of x. We will then plot all of these points on a rectangular coordinate system, sketch the graph which they form, and try to guess the function which has this graph. For each value of x that is entered into the program, it will give as output x, sin(x) and values of h and $\frac{\sin(x+h)-\sin(x)}{h}$ for h=±.1, ±.01, ±.001, and ±.0001.

Problem 14: Use the program /DERIVSIN/ to find the derivative function of sin(x). Select values for x until you have enough approximations to the derivative of sin(x) to sketch reasonably good graphs of both sin(x) and D_x sin(x). What is your guess as to the derivative function of sin(x)?

Problem 15: Write a program similar to /DERIVSIN/ which for any value of x will give as output

$$x, \quad e^x$$
$$\text{and } h, \quad \frac{e^{x+h}-e^x}{h}$$

for h=±.1, ±.01, ±.001, and ±.0001.

Use this program to sketch graphs of $y=e^x$ and the derivative function of e^x.

Problem 16: Write a program which for any value of x will give as output x, cos(x) and h, $\frac{\cos(x+h)-\cos(x)}{h}$

for h=±.1, ±.01, ±.001, and ±.0001.

Use this program to discover the derivative function of cos(x).

A sample run of the program /DERIVSIN/.

```
PRØGRAM TØ ESTIMATE THE DERIVATIVE ØF
SIN(X) FØR ANY VALUE ØF X.

WHAT IS X?
?-1


X =    -1     SIN(X) =     -0.841470985

     H            [SIN(X+H)-SIN(X)]/H

 0.1              0.581440752
-0.1              0.497363753
 9.999999999E-03                0.54450062
-9.999999999E-03                0.536085982
 9.999999999E-04                0.540722944
-9.999999999E-04                0.53988149
 9.999999998E-05                0.540344336
-9.999999998E-05                0.540260335

WHAT IS X?
?0


X =     0     SIN(X) =      0

     H            [SIN(X+H)-SIN(X)]/H

 0.1              0.998334166
-0.1              0.998334166
 9.999999999E-03                0.999983333
-9.999999999E-03                0.999983333
 9.999999999E-04                0.999999833
-9.999999999E-04                0.999999833
 9.999999998E-05                1
-9.999999998E-05                1

WHAT IS X?
?1


X =     1     SIN(X) =      0.841470985

     H            [SIN(X+H)-SIN(X)]/H

 0.1              0.497363753
-0.1              0.581440752
 9.999999999E-03                0.536085981
-9.999999999E-03                0.544500621
 9.999999999E-04                0.539881479
-9.999999999E-04                0.540722951
 9.999999998E-05                0.540260226
-9.999999998E-05                0.540344372

WHAT IS X?
?←←ESC:          16
```

A listing of the program /DERIVSIN/.

```
10 PR. "PRØGRAM TØ ESTIMATE THE DERIVATIVE ØF"
12 PR. "SIN(X) FØR ANY VALUE ØF X."
14 PR.
16 PR. "WHAT IS X?" INPUT X
18 PR.
20 PR.
30 LET Z=SIN(X)
40 PR. "X = ";X;"SIN(X) = ";Z
42 PR.
50 PR."     H          [SIN(X+H)-SIN(X)]/H"
52 PR.
60 FØR N=1 TØ 4
70 LET H1 = 10.0↑(-N)
80 LET H2 = -H1
90 LET Y1=(SIN(X+H1)-Z)/H1
100 LET Y2=(SIN(X+H2)-Z)/H2
110 PR. H1, Y1
120 PR. H2, Y2
130 NEXT N
140 GØTØ 14
```

The following program is a generalization of the program /DERIVSIN/ which can be used to explore the derivative function of sin(x). When using this program, you have the option of selecting any function whose derivative you want to study. To use this program called /DERIVTIV/, first load the function into your workspace. Don't try to run this program until you alter it by providing the function whose derivative you are going to study. You can do this by typing

```
>100 LET Y=F(X)
```

where F(X) is any function you want. For example to find approximations to values of the derivative function of $y=\sqrt{x+4}$ you must type

```
>100 LET Y=SQRT(X+4.0)
```

Of course you must remember that x must be greater than or equal to -4 or SQRT(X+4.0) will be imaginary.

When you finish using the program to explore your function, hit the ESCAPE key. Now you can either LOGOUT or work on another function y=G(X) by typing

```
>100 LET Y=G(X)
>RUN
```

where G(X) is a specified function of X.

The following pages contain a set of sample output from /DERIVTIV/ and a listing of the program.

Problem 17: Use /DERIVTIV/ to obtain enough data to plot an accurate graph of $y=\frac{1}{x}$ and the derivative function of $y=\frac{1}{x}$. Plot these graphs and try to guess the function which is the derivative of $y=\frac{1}{x}$.

Problem 18: Use /DERIVTIV/ to assist in plotting the graphs of $y=|x|$ and its derivative function. Try to guess the derivative function of $y=|x|$.

Problem 19: Use /DERIVTIV/ to assist in plotting the graphs of y=sin(2x) and the derivative function of y=sin(2x). Can you guess what this derivative function is?

A sample run of the program /DERIVTIV/.

```
>100 LET Y=EXP(X)

>RUN

THIS PRØGRAM WILL CØMPUTE A SET ØF APPRØXIMATIØNS
TØ THE DERIVATIVE ØF ANY FUNCTIØN F(X) WHICH YØU GIVE
IT FØR ANY VALUE YØU ASSIGN TØ THE INDEPENDENT
VARIABLE X, PRØVIDED THE FUNCTIØN HAS A DERIVATIVE
FØR THAT X.

FØR EACH X YØU WILL GET AS ØUTPUT THE VALUES ØF X
AND F(X), AS WELL AS A SET ØF APPRØXIMATIØNS TØ
F'(X) FØR H = .1, -.1, .01, -.01, .001, -.001,
.0001, AND - .0001.

WHAT IS X?
?0

     H       APPRØXIMATIØN TØ F'(X)
 0.1              1.051709181
-0.1              0.95162582

 9.999999999E-03                    1.005016708
-9.999999999E-03                    0.995016639

 9.999999999E-04                    1.000500161
-9.999999999E-04                    0.999500247

 9.999999998E-05                    1.000049961
-9.999999998E-05                    0.999950571


X =    0     Y =      1
WHAT IS X?
?1

     H       APPRØXIMATIØN TØ F'(X)
 0.1             2.858841955
-0.1             2.586787173

 9.999999999E-03                    2.73191865
-9.999999999E-03                    2.704735608

 9.999999999E-04                    2.71964143
-9.999999999E-04                    2.716923176

 9.999999998E-05                    2.718417673
-9.999999998E-05                    2.71814628


X =    1     Y =      2.718281828
WHAT IS X?
?←ESC:
```

A listing of the program /DERIVTIV/.

```
4 PR. "THIS PRØGRAM WILL CØMPUTE A SET ØF APPRØXIMATIØNS"
6PR. "TØ THE DERIVATIVE ØF ANY FUNCTIØN F(X) WHICH YØU GIVE"
8 PR. "IT FØR ANY VALUE YØU ASSIGN TØ THE INDEPENDENT"
10 PR. "VARIABLE X, PRØVIDED THE FUNCTIØN HAS A DERIVATIVE"
12 PR. "FØR THAT X."
14 PR.
16 PR. "FØR EACH X YØU WILL GET AS ØUTPUT THE VALUES ØF X"
18 PR. "AND F(X), AS WELL AS A SET ØF APPRØXIMATIØNS TØ"
20 PR. "F'(X) FØR H = .1, -.1, .01, -.01, .001, -.001,"
22 PR. ".0001, AND - .0001."
24 PR.
25 PR.
26PR.
28 PR. "WHAT IS X?" INPUT X
30 PR.
32 PR. "     H      APPRØXIMATIØN TØ F'(X)"
35 FØR N=1 TØ 4
40 LET H=10↑(-N)
45 LET C = 0
47 LET I=0
50 LET A=X
55 LET X=X+H
100 LET Y=EXP(X)
110 IF I=1 GØTØ 200
120 LET I=I+1
130 LET D1=Y
140 LET X=A
150 GØTØ 100
200 LET D2=Y
210 LET D=(D1-D2)/H
220 PR. H, D
230 LET C=C+1
240 IF C=2 GØTØ 270
250 LET H = (-H)
260 GØTØ 47
270 PR.
300 NEXT N
320 PR.
330 PR.
340 PR.        "X = ";A; " Y = ";D2
350 GØTØ 25
```

THE FUNDAMENTAL THEOREM OF INTEGRAL CALCULUS

A modification of /INTEGRATE/ can be used to "numerically verify" The Fundamental Theorem of Integral Calculus; i.e., $\int_a^b f(x)dx=F(b)-F(a)$ where F(x) is a primitive of f(x).

For example,

$$\int_0^{\pi/2} \sin(x)dx=\left[-\cos(\pi/2)\right]-\left[-\cos(0)\right]$$

can be verified by calculating a sequence of approximations to $\int_0^{\pi/2} \sin(x)dx$ and comparing the suspected limit of the sequence to $\left[-\cos(\pi/2)\right]-\left[-\cos(0)\right]=1$.

The following program, /FUNDTHEO/, uses the trapezoid method to compute approximations to $\int_a^b f(x)dx$ for a given function f(x) over a specified interval [a, b] for n trapezoids, and compares the approximation to g(b)-g(a) where g(x) is a primitive of f(x).

After you have loaded the program, modify lines 100 and 200 as follows:

Line 100 should contain the function to be integrated, Y=F(X) in the form:

```
>100 LET Y=F(X)
```

Line 200 should contain any primitive Z=G(X) of Y=F(X) in the form:

```
>200 LET Z=G(X)
```

For example, if the integral is $\int_a^b \cos(x)dx$, then line 100 should be:

```
>100 LET Y=COS(X)
```

and line 200 could be:

```
>200 LET Z=SIN(X)+1
```

since SIN(X)+1 is a primitive of COS(X).

After you have supplied lines 100 and 200 you can execute the program by typing

```
>RUN
```

After the program has printed an approximation to $\int_a^b f(x)dx$ and G(B)-G(A) it will print

TYPE THE CODE

To use a different number of approximating trapezoids on the interval [a, b] type 30; to get a new interval by changing the values of a and b, type 20; to stop the program type 300.

<u>Problem 20</u>: Use /FUNDTHEO/ to approximate $\int_0^2 -x^2 dx$ and to verify that $\int_0^2 -x^2 dx = \left[-\frac{x^3}{3}+2\right]_0^2$.

<u>Problem 21</u>: Use /FUNDTHEO/ to approximate $\int_0^{\pi/2} \cos^2(x)dx$ and to verify that $\int_0^{\pi/2} \cos^2(x)dx = \left[\frac{x}{2} + \frac{\sin(x)\cos(x)}{2} - 1\right]_0^{\pi 2}$

<u>Problem 22</u>: Use /FUNDTHEO/ to verify $\int_0^1 \frac{dx}{x^2+1} = \left[\arctan(x)\right]_0^1$

<u>Problem 23</u>: Use /FUNDTHEO/ to verify

A. $\int_2^4 e^{3x}dx = \left[\frac{e^{3x}}{3} - 6\right]_2^4$

B. $\int_2^3 \log_e(x)dx = \left[x(\log_e x) - x\right]_2^3$

C. $\int_{\frac{1}{2}}^1 \sec^2(x)dx = \left[\tan(x) - 2\right]_{\frac{1}{2}}^1$

D. $\int_{-1}^1 \sqrt{1+\cos(x)}\ dx = \left[2\sqrt{2}\ (\sin\frac{x}{2})\right]_{-1}^1$

A sample run of the program /FUNDTHEO/.

```
>RUN
THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE DEFINITE
INTEGRAL ØF THE FUNCTIØN Y = F(X) WHICH YØU SUPPLIED ØN
LINE 100, ØVER AN INTERVAL [A,B]; AND CØMPARES THE
APPRØXIMATIØN TØ G(B)-G(A), WHERE Z = G(X) IS A
PRIMITIVE ØF F(X), WHICH YØU SUPPLIED ØN LINE 200.

WHAT IS A?
?-3.141529
INPUT B.
?0
HØW MANY SUBINTERVALS ØN [A,B]?
?200
INTERVAL [  -3.141529   ,   0    ].     200  TRAPEZØIDS

APPRØXIMATIØN TØ INTEGRAL IS *******   -1.999958877   *******

G(B) - G(A) IS    -1    -   0.999999998   =  -1.999999998


TYPE THE CØDE
?20

WHAT IS A?
?0
INPUT B.
?3.141592
HØW MANY SUBINTERVALS ØN [A,B]?
?200
INTERVAL [   0    ,   3.141592   ].     200  TRAPEZØIDS

APPRØXIMATIØN TØ INTEGRAL IS *******    1.999958882   *******

G(B) - G(A) IS     1    -   -1    =   2


TYPE THE CØDE
?20

WHAT IS A?
?0
INPUT B.
?10
HØW MANY SUBINTERVALS ØN [A,B]?
?200
INTERVAL [   0    ,   10    ].     200  TRAPEZØIDS

APPRØXIMATIØN TØ INTEGRAL IS *******    1.825087846   *******

G(B) - G(A) IS    0.839071529   -  -1    =   1.839071529


TYPE THE CØDE
?300
```

A listing of the program /FUNDTHEO/.

```
10PR."THIS PRØGRAM CØMPUTES APPRØXIMATIØNS TØ THE DEFINITE"
12PR."INTEGRAL ØF THE FUNCTIØN Y = F(X) WHICH YØU SUPPLIED ØN "
14PR."LINE 100, ØVER AN INTERVAL [A,B]; AND CØMPARES THE "
16PR."APPRØXIMATIØN TØ G(B)-G(A), WHERE Z = G(X) IS A"
18PR."PRIMITIVE ØF F(X), WHICH YØU SUPPLIED ØN LINE 200."
20PR.
22PR."WHAT IS A?" INPUT A
24PR."INPUT B." INPUT B
26 IF A>=B GØTØ 296
30 PR. "HØW MANY SUBINTERVALS ØN [A,B]?"
31 INPUT N
32 LET N = INT(ABS(N))
40 LET S=0
50 LET W=(B-A)/N
60 FØR X=A TØ B STEP W
100 LET Y = SIN(X)
105 IF X=B GØTØ 125
110 IF X=A GØTØ 125
115 LET S=S+(2*Y)
120 GØTØ 130
125 LET S = S+Y
130 NEXT X
135 LET T=((B-A)/(2*N))*S
140PR."INTERVAL ["；A；","；B；"]."；N；"TRAPEZØIDS"
142 PR.
144PR."APPRØXIMATIØN TØ INTEGRAL IS *******"；T；"*******"
146PR.
195 LET L=0
197 LET X=A
200 LET Z=-CØS(X)
205 IF L=1 GØTØ 250
210 LET L=L+1
215 LET C=Z
220 LET X = B
225 GØTØ 200
250 LET D=Z
255 LET R=D-C
260 PR."G(B) - G(A) IS"；D；"-"；C；"="；R
265 PR.
270 PR.
275 PR."TYPE THE CØDE" INPUT E
280 IF E = 20 GØTØ 20
285 IF E=30 GØTØ 30
290 IF E=300 GØTØ 300
292 PR."THE CØDE MUST BE 20, 30, ØR 300."
294 GØTØ 275
296 PR."A MUST BE LESS THAN B."
298 GØTØ 22
300 END
```

SKETCHING GRAPHS OF FUNCTIONS

Constructing the Graph of a Function

In this section we are concerned with methods for sketching graphs of functions. One approach to sketching a graph of the function $y = f(x)$ is to select various values for x, to calculate the corresponding values for y by substituting values of x into the function $f(x)$, and to plot the set of ordered pairs of numbers $\{(x, f(x))\}$ on rectangular coordinate paper, using an appropriate scale for the x and y axes. When enough ordered pairs have been calculated and plotted to enable you to predict the shape of the graph with reasonable accuracy, the points can be connected by smooth lines (not necessarily straight lines) to obtain a graph of the function $y = f(x)$. If the graph has been precisely constructed, it is possible to obtain much information about the function by studying the graph.

When analyzing a function, one should consider the behavior of the function relative to any open or closed interval in the domain of the function with respect to the following mathematical concepts:

(1) Is the function increasing or decreasing?
(2) Is the function one-to-one?
(3) Is the function onto?
(4) Does the function have an inverse function?
(5) Does the function have any real zeros (roots)?
(6) Is the function strictly increasing or strictly decreasing?
(7) Is the function constant?
(8) Does the function have maximum or minimum values?
(9) Does the function have relative maximum or relative minimum values?
(10) Is the function concave upward or concave downward?
(11) Are there any real numbers for which the function is undefined?
(12) Are there any real numbers for which the function is discontinuous?

(13) Are there any real numbers for which the function does not have a derivative?
(14) What is the range of the function?
(15) What is the domain of the function?

Check your sources of information (books, other students, teachers, etc.) to make certain that you understand each mathematical term that is underlined.

Problem 24: Write computer programs which will accept as input numerical values of x and give as output values of f(x) for each of the following functions. Use your program to obtain enough points on the graph of each function so that you can plot an accurate graph of the function.

A. $f(x) = x^2-3x+2$

B. $f(x) = -x^2+3x-2$

C. $f(x) = \cos(x)$

D. $f(x) = 2\cos(x)$

E. $f(x) = \cos(2x)$

F. $f(x) = \log_4 x$

G. $f(x) = \begin{cases} \sin(\frac{1}{x}) & \text{if } x\neq 0 \\ 0 & \text{if } x=0 \end{cases}$

H. $f(x) = x(\sin(x))$

I. $f(x) = \frac{e^x}{x}$

This module is a guessing game. The program called /GRAPH1/ contains five different functions of x. For any of the functions and for any real value of x which you select, the program will give as output:

(1) that value of x,
(2) the value of the function,
(3) the value of the derivative of the function,
(4) the value of the second derivative of the function.

Your job is to select as many of these functions as you want, sketch graphs of the functions and their first and second derivative functions, and try to guess what the functions are and what the derivative functions are. To sketch the graphs you will have to make shrewd selections for your values of x in order to get points which will assist you in obtaining accurate graphs.

You should try to find maximum and minimum points on the graphs, points where the functions are discontinuous or undefined, intervals where the functions are either increasing or decreasing, intervals where the functions are either concave upward or downward and points where the graphs cross the x and y axes.

After you have loaded the program /GRAPH1/ and typed:

```
>RUN
```

you will get the message

```
NOW TYPE THE CODE NUMBER OF THE FUNCTION
THAT YOU WANT TO WORK WITH.
?
```

The code numbers of the functions are 100, 200, 300, 400, and 500; so you can select a function by typing any one of these numbers following the question mark. After you have selected a function you will be asked to type a value of x. When you supply the value of x you will ob-

tain as output that value of x, together with the values of the function and its first and second derivatives. The message:

TYPE A VALUE FOR X

will be repeated five times, then you will have the option of getting values of the function and its derivatives for five more values of x or of going on to another function.

A sample output from this program is shown on the next page.

A sample run of the program /GRAPH1/. 6.5

```
NØW TYPE THE CØDE NUMBER ØF THE FUNCTIØN
THAT YØU WANT TØ WØRK WITH.
?100
TYPE A VALUE FØR X.
?-2
X =   -2     Y =        4     D1 =      -4     D2 =      2
TYPE A VALUE FØR X.
?-1
X =   -1     Y =        1     D1 =      -2     D2 =      2
TYPE A VALUE FØR X.
?0
X =    0     Y =        0     D1 =       0     D2 =      2
TYPE A VALUE FØR X.
?1
X =    1     Y =        1     D1 =       2     D2 =      2
TYPE A VALUE FØR X.
?
 2
X =    2     Y =        4     D1 =       4     D2 =      2
IF YØU HAVE ENØUGH INFØRMATIØN ABØUT THIS FUNCTIØN,
TYPE STØP; FØR MØRE INFØRMATIØN, TYPE GØ.
?STØP
TØ ØBTAIN ACCESS TØ ANØTHER FUNCTIØN TYPE ITS CØDE
NUMBER, 100, 200, 300, 400, ØR 500.  TØ STØP THE
PRØGRAM, HIT THE ESCAPE KEY.
NØW TYPE THE CØDE NUMBER ØF THE FUNCTIØN
THAT YØU WANT TØ WØRK WITH.
?400
TYPE A VALUE FØR X.
?-5
X =   -5     Y =      0.958924275    D1 =      0.283662185    D2 =
-0.958924275
TYPE A VALUE FØR X.
?
 -2
X =   -2     Y =     -0.909297427    D1 =     -0.416146837    D2 =
 0.909297427
TYPE A VALUE FØR X.
?0
X =    0     Y =        0     D1 =       1     D2 =      0
TYPE A VALUE FØR X.
?1
X =    1     Y =      0.841470985    D1 =      0.540302306    D2 =
-0.841470985
TYPE A VALUE FØR X.
?10
X =   10     Y =     -0.544021111    D1 =     -0.839071529    D2 =
 0.544021111
IF YØU HAVE ENØUGH INFØRMATIØN ABØUT THIS FUNCTIØN,
TYPE STØP; IF YØU WANT MØRE INFØRMATIØN, TYPE GØ.
?STØP
TØ ØBTAIN ACCESS TØ ANØTHER FUNCTIØN TYPE ITS CØDE
NUMBER, 100, 200, 300, 400, ØR 500.  TØ STØP THE
PRØGRAM, HIT THE ESCAPE KEY.
NØW TYPE THE CØDE NUMBER ØF THE FUNCTIØN
THAT YØU WANT TØ WØRK WITH.
?300
TYPE A VALUE FØR X.
?-4
X =   -4     Y =      0.018315639    D1 =      0.018315639    D2 =
 0.018315639
TYPE A VALUE FØR X.
?←ESC:          305
```

A listing of the program /GRAPH1/.


```
34 PRINT " "
36 PRINT "NOW TYPE THE CODE NUMBER OF THE FUNCTION"
38 PRINT "THAT YOU WANT TO WORK WITH."
44 INPUT C
46 IF C = 100 GOTO 100
48 IF C = 200 GOTO 200
50 IF C = 300 GOTO 300
52 IF C = 400 GOTO 400
54 IF C = 500 GOTO 500
56 PRINT "YOU MUST TYPE 100, 200, 300, 400, OR 500."
58 PRINT "TRY AGAIN."
60 GOTO 44
100 LET N = 0
102 PRINT "TYPE A VALUE FOR X."
105 INPUT X
107 LET Y=X*X
110 LET D1=2*X
115 LET D2=2
120 PRINT "X = ";X;" Y = ";Y;"D1 = ";D1;" D2 = ";D2
125 LET N = N+1
130 IF N=5 GOTO 150
135 GOTO 102
150 PRINT "IF YOU HAVE ENOUGH INFORMATION ABOUT THIS FUNCTION,"
155 PRINT "TYPE STOP; FOR MORE INFORMATION, TYPE GO."
165 INPUT T$
170 IF IS (T$,"GO",0) GOTO 100
175 IF IS (T$,"STOP",0) GOTO 600
180 PRINT "YOU MUST TYPE STOP OR GO.  TRY AGAIN."
185 GOTO 165
200 LET N=0
202 PRINT "TYPE A VALUE FOR X."
205 INPUT X
207 LET Y=X*(X-4)
210 LET D1=2*(X-2)
215 LET D2=2
220 PRINT "X = ";X;" Y = ";Y;" D1 = ";D1;" D2 = ";D2
225 LET N=N+1
230 IF N = 5 GOTO 250
235 GOTO 202
250 PRINT "IF YOU HAVE ENOUGH INFORMATION ABOUT THIS FUNCTION,"
255 PRINT "TYPE STOP;  IF YOU WANT MORE INFORMATION, TYPE GO."
260 INPUT T$
270 IF IS (T$,"GO",0) GOTO 200
275 IF IS (T$,"STOP",0) GOTO 600
280 PRINT " YOU  MUST TYPE STOP OR GO.  TRY AGAIN."
285 GOTO 260
300 LET N = 0
302 PRINT "TYPE A VALUE FOR X."
305 INPUT X
307 LET Y = EXP(X)
310 LET D1=Y
315  LET D2=Y
320 PRINT "X = ";X;" Y = ";Y;" D1 = ";D1;" D2 = ";D2
325 LET N = N+1
330 IF N = 5 GOTO 350
335 GOTO 302
350 PRINT " IF YOU HAVE ENOUGH INFORMATION ABOUT THIS FUNCTION,"
355 PRINT "TYPE STOP; IF YOU WANT MORE INFORMATION, TYPE GO."
360 INPUT T$
370 IF IS (T$,"GO",0) GOTO 300
375 IF IS (T$,"STOP",0) GOTO 600
380 PRINT " YOU MUST TYPE STOP OR GO. TRY AGAIN."
385 GOTO 360
```

A listing of the program /GRAPH1/ (continued).

```
400 LET N=0
402 PRINT "TYPE A VALUE FØR X."
405 INPUT X
407 LET Y = SIN(X)
410 LET D1=CØS(X)
415 LET 02 =-SIN(X)
420 PRINT "X = ";X;" Y = ";Y;" 01 = ";01;" 02 = ";02
425 LET N = N+1
430 IF N = 5 GØTØ 450
435 GØTØ 402
450 PRINT "IF YØU HAVE ENØUGH INFØRMATIØN ABØUT THIS FUNCTIØN,"
455 PRINT "TYPE STØP; IF YØU WANT MØRE INFØRMATIØN, TYPE GØ."
460 INPUT T$
470 IF IS (T$,"GØ",0) GØTØ 400
475 IF IS (T$,"STØP",0) GØTØ 600
480 PRINT " YØU MUST TYPE STØP ØR GØ. TRY AGAIN."
485 GØTØ 460
500 LET N=0
502 PRINT "TYPE A VALUE FØR X."
505 INPUT X
507 LET Y = X*(X*X-X+1)
510 LET 01=X*(3*X-2)+1
515 LET 02=6*X-2
520  PRINT "X = ";X;" Y = ";Y;" 01 = ";01;" 02 = ";02
525 LET N=N+1
530 IF N = 5 GØTØ 550
535 GØTØ 502
550 PRINT "IF YØU HAVE ENØUGH INFØRMATIØN ABØUT THIS FUNCTIØN,"
555 PRINT "TYPE STØP; IF YØU WANT MØRE INFØRMATIØN,TYPE GØ."
560 INPUT T$
570 IF IS (T$,"GØ",0) GØTØ 500
575 IF IS(T$,"STØP",0) GØTØ 600
580 PRINT " YØU MUST TYPE STØP ØR GØ.  TRY AGAIN."
585 GØTØ 560
600 PRINT "TØ ØBTAIN ACCESS TØ ANØTHER FUNCTIØN TYPE ITS CØOE"
605 PRINT "NUMBER, 100, 200, 300, 400, ØR 500.  TØ STØP THE"
610 PRINT "PRØGRAM, HIT THE ESCAPE KEY."
650 GØTØ 36
```

This module called /GRAPH2/ is a modification of /GRAPH1/. It is a game for guessing functions; however your guesses must be based upon less information than is given as output from /GRAPH1/. The program /GRAPH2/ contains the first and second derivative functions for six different functions of x. For any of these six functions and for any real value of x which you select, the program will give you for output:

(1) the value of x which you selected,
(2) the value of the derivative of the function,
(3) the value of the second derivative of the function.

Your problem is to select some or all of the six functions, to get as much information about the first and second derivatives of the functions as you need in order to sketch accurate graphs of the derivative functions, and finally to guess the functions whose first and second derivative functions are stored in the computer.

You should review the information concerning the relationship of a function to its derivative function, and be especially concerned with maximum and minimum points, inflection points, critical points, and intervals where the function is increasing or decreasing and concave upward or downward.

After you have loaded /GRAPH2/ and typed:

```
>RUN
```

you will get the message

```
NOW SELECT A FUNCTION BY TYPING ITS CODE
NUMBER; 100, 200, 300, 400, 500, OR 600.
?
```

You can now select any of the six functions by typing its code number following the question mark. After you have chosen a function to study, you will be asked to type a value for x. After typing a value for x you will be given as output that value of x together with corresponding values of the first and second derivative functions of the function which you are attempting to guess.

The message

TYPE THE VALUE FOR X.

will be repeated five times, then you will have the option of getting values of the derivatives for five more values of x or of going on to another function. When you have enough information about the functions, hit the escape key.

A sample output from this program is shown on the next page.

A sample output of the program /GRAPH2/.


```
NØW SELECT A FUNCTIØN BY TYPING ITS CØDE
NUMBER: 100, 200, 300, 400, 500, ØR 600.
?200
TYPE THE VALUE ØF X.
?-1
X=    -1     F1=     0.367879441     F2=     0.367879441
TYPE THE VALUE ØF X.
?0
X=     0     F1=     1     F2=     1
TYPE THE VALUE ØF X.
?2
X=     2     F1=     7.389056099     F2=     7.389056099
TYPE THE VALUE ØF X.
?3
X=     3     F1=     20.08553692     F2=     20.08553692
TYPE THE VALUE ØF X.
?-2
X=    -2     F1=     0.135335283     F2=     0.135335283
TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ.
TØ LEAVE THIS FUNCTIØN, TYPE STØP.
?GO
TYPE THE VALUE ØF X.
?1.5
X=     1.5   F1=     4.48168907      F2=     4.48168907
TYPE THE VALUE ØF X.
?2
X=     2     F1=     7.389056099     F2=     7.389056099
TYPE THE VALUE ØF X.
?8
X=     8     F1=     2980.957987     F2=     2980.957987
TYPE THE VALUE ØF X.
?-9
X=    -9     F1=     1.234098041E-04   F2=     1.234098041E-04
TYPE THE VALUE ØF X.
?7
X=     7     F1=     1096.633158     F2=     1096.633158
TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ.
TØ LEAVE THIS FUNCTIØN, TYPE STØP.
?STP
TYPE EITHER STØP ØR GØ. TRY AGAIN.
?STØP
TØ GET ANØTHER FUNCTIØN TYPE ITS CØDE NUMBER.
TØ STØP THE PRØGRAM HIT THE ESCAPE KEY.
?300
TYPE A VALUE FØR X.
?-1
X=    -1     F1=    -1     F2=    0
TYPE A VALUE FØR X.
?0
X=0, F1 AND F2 DØ NØT EXIST.
TYPE A VALUE FØR X.
?3
X=     3     F1=     1     F2=    0
TYPE A VALUE FØR X.
?5
X=     5     F1=     1     F2=    0
TYPE A VALUE FØR X.
?8
X=     8     F1=     1     F2=    0
TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ.
TØ LEAVE THIS FUNCTIØN, TYPE STØP.
?←←←←ESC:          510
```

A listing of the program /GRAPH2/.

```
46 PR. " "
48 PR. "NØW SELECT A FUNCTIØN BY TYPING ITS CØDE"
50 PR. "NUMBER; 100, 200, 300, 400, 500, ØR 600."
52 INPUT C
54 IF C = 100 GØTØ 100
56 IF C = 200 GØTØ 200
58 IF C = 300 GØTØ 300
60 IF C = 400 GØTØ 400
62 IF C = 500 GØTØ 500
64 IF C = 600 GØTØ 600
66 PR. "YØU MUST TYPE EITHER 100, 200, 300, 400,"
68 PR. "500, ØR 600.  TRY AGAIN."
70 GØTØ 52
100 LET N = 0
105 PR. "NØW TYPE A VALUE FØR X."
110 INPUT X
115 LET F1=2
120 LET F2=0
125 PR. "X = ";X;" F1 = ";F1;" F2 = ";F2
130 LET N = N+1
135 IF N=5 GØTØ 150
140 GØTØ 105
150 PR. "TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ."
155 PR. "TØ GET ANØTHER FUNCTIØN ØF X, TYPE STØP."
160 INPUT T$
165 IF IS (T$,"GØ",0) GØTØ 100
170 IF IS (T$,"STØP",0) GØTØ 700
175 PR. "YØU MUST TYPE GØ ØR STØP. TRY AGAIN."
180 GØTØ 160
200 LET N=0
205 PR. "TYPE THE VALUE ØF X."
210 INPUT X
215 LET F1=EXP(X)
220 LET F2=F1
225 PR. "X= ";X;" F1= ";F1;" F2= ";F2
230 LET N=N+1
235 IF N=5 GØTØ  250
240 GØTØ 205
250 PR. "TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ."
255 PR. "TØ LEAVE THIS FUNCTIØN, TYPE STØP."
260 INPUT T$
265 IF IS (T$,"GØ",0) GØTØ 200
270 IF IS (T$,"STØP",0) GØTØ 700
275 PR. "TYPE EITHER STØP ØR GØ. TRY AGAIN."
280 GØTØ 260
300 LET N=0
305 PR. "TYPE A VALUE FØR X."
310 INPUT X
315 IF X=0 GØTØ 385
320 IF X<0 LET F1=-1
325 IF X>0 LET F1=1
330 LET F2=0
335 PR. "X= ";X;" F1= ";F1;"F2= ";F2
340 LET N = N+1
350 IF N=5 GØTØ 360
355 GØTØ 305
```

A listing of the program /GRAPH2/ (continued)

```
360 PR. "TØ ASSIGN 5 MØRE VALUES TØ X, TYPE GØ."
365 PR. " TØ LEAVE THIS FUNCTIØN, TYPE STØP."
370 INPUT T$
372 IF IS (T$,"GØ",0) GØTØ 300
374 IF IS (T$,"STØP",0) GØTØ 700
376 PR. "TYPE EITHER GØ ØR STØP.  TRY AGAIN."
380 GØTØ 370
385 PR. "X=0, F1 AND F2 DØ NØT EXIST."
390 GØTØ 340
400 LET N=0
405 PR. "TYPE A VALUE FØR X."
410 INPUT X
415 LET F1=-SIN(X)
420 LET F2=-CØS(X)
425 PR. "X= ";X;" F1= ";F1;"F2= ";F2
430 LET N=N+1
435 IF N=5 GØTØ 450
440 GØTØ 405
450 PR. "TØ GIVE X FIVE MØRE VALUES, TYPE GØ."
455 PR. "TØ GET ANØTHER FUNCTIØN, TYPE STØP."
460 INPUT T$
465 IF IS (T$,"GØ",0) GØTØ 400
470 IF IS (T$,"STØP",0) GØTØ 700
475 PR. " TYPE STØP ØR GØ. TRY AGAIN."
480 GØTØ 460
500 LET N=0
505 PR. "TYPE A VALUE FØR X."
510 INPUT X
515 LET F1=3*X*X
520 LET F2=6*X
525 PR. "X= ";X;" F1= ";F1;"F2= ";F2
530 LET N=N+1
535 IF N=5 GØTØ 550
540 GØTØ 505
550 PR. "TØ GIVE X MØRE VALUES, TYPE GØ."
555 PR. " TØ GØ TØ A DIFFERENT FUNCTIØN, TYPE STØP."
560 INPUT T$
565 IF IS (T$,"GØ",0) GØTØ 500
570 IF IS (T$,"STØP",0) GØTØ 700
575 PR. "TYPE EITHER STØP ØR GØ.  TRY AGAIN."
580 GØTØ 560
600 LET N=0
605 PR. "TYPE A VALUE FØR X."
610 INPUT X
615 LET F1=X*(X-3)+2
620 LET F2=2*X-3
625 PR. "X= ";X;" F1= ";F1;"F2 = ";F2
630 LET N=N+1
635 IF N=5 GØTØ 650
640 GØTØ 605
650 PR. "TØ GIVE MØRE VALUES TØ X, TYPE GØ."
655 PR. "TØ GØ TØ ANØTHER FUNCTIØN, TYPE STØP."
660 INPUT T$
665 IF IS (T$,"GØ",0) GØTØ 600
670 IF IS (T$,"STØP",0) GØTØ 700
675 PR. "YØU MUST TYPE GØ ØR STØP. TRY AGAIN."
680 GØTØ 660
700 PR. "TØ GET ANØTHER FUNCTIØN TYPE ITS CØDE NUMBER."
705 PR. "TØ STØP THE PRØGRAM HIT THE ESCAPE KEY."
710 GØTØ 52
```